Technical University of Cluj Napoca

Faculty of Automation and Computer Science

Homework 5 – Dictionary

Lorand Berekmeri

Group: 30425

Date: 20.05.2016

Objective

The objective of this theme is to design and implement a dictionary of synonyms in Romanian. Specifically, we have more functions specific to online dictionaries. Such as adding a word synonymous with his updating a synonym Dictionary, delete a word, or we can look for words in the dictionary after a few letters that make the word (using shortcuts "\*?").

To stop using bases DATEM this time we use a file to store data equiped. So we run the app take over all the data in the file and at the end they all application data in the file we will write back. Another II Project of the issue is understanding the concepts of serializable and deserializable objects. Another goal is the same like the last project, to get familiarize with Design by Contract programing.

Task

First, we need to solve the theme of a good grasp of the principle serializer / deserializer. Object serialization means saving and restoring state objects. Objects of any class that implements the interface Serializable can be saved to a stream (fluxde data) and restored it. Java.io package contains two special classes for serialization ObjectOutputStream ObjectInputStream respectively primitive types.

Desing by Contract is a "contract" that specifies restrictions that must obey input from one method, the possible values ​​of output and moods where you can find program - these restrictions are given in the form of: a) Prerequisite is obligations that input of a method must comply with for the method to work properly b) postconditions: a guarantee that the output data from one method gives c) invariably represent conditions imposed on states in the program may be at a time

The application must achieve an interface with utlilizatorul and dictionary. The user must can add new words with synonyms, to amend certain synonyms or remove some words in the dictionary. Thus all the data from the application must be retained in a file, and each time you open the application all data in the file will automatically be entered in our application, as application and exit all data will be written back into the file.

Modelling

In achieving modeling application we used the concept of Desing but by Contract, Serializable and Deserializable. Word - synonym for retention, I used an object HashMap objects to associate each item a unique key (key pairs type - value). Therefore can not contain duplicate keys and each key is associated to a single element. Thus, this HashMap is as key a String, representing another word and String representing synonym keyword. We used the concept of Design by Contract by implementing DictionaryProc, Dictionary class implementing this interface. In Regarding the OOP paradigm, all the attributes of all classes are private, so encapsulation performes in every class. I used inheritance and Design by Contract.

How to use the program

To use this program the user must follow the following string

events:

1.The user running the

2. Choose the command you want to do:

Visualization Dictionary

-append words

-updating synonyms

Deletes words

-Search by \* or searching by whole word

3. The user enters data (word -synonym).

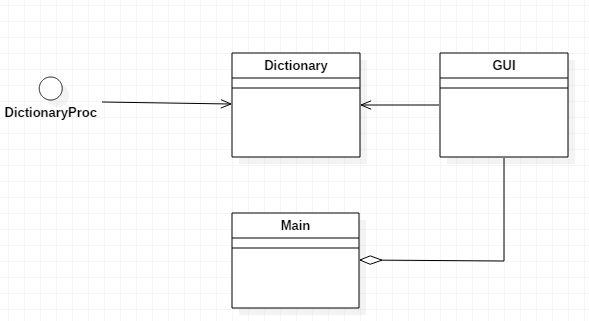
4. consumers the corresponding button press operation that executes wants to await

the surgery to be performed.

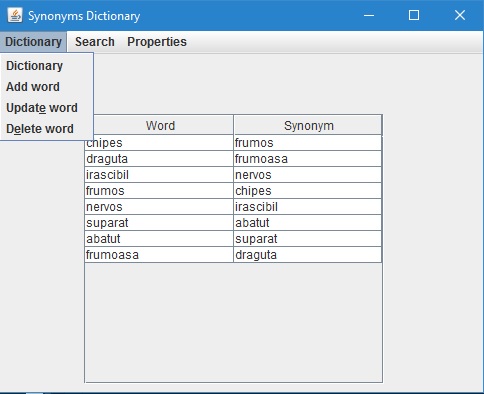
5. execute the program and displays the result.

6. To exit the application press the Exit button. (X)

*Class Diagram*



*Interface*



*GUI*

Graphical User Interface (GUI) is a term understood And that refers to all types of visual communication between a program and its users, this is a customization of the user interface (UI), by which we understand the general concept of interaction between Program and users. Java offers many classes for implementing various UI features.

In principle, the creation of a grafuce applications includes:  
-A Design: Creating a viewing area that will be placed graphic objects (components), to communicate with the user buttons, controls for editing texts, lists, etc.), creation and placement of components on the surface of display positions corespunsatoare.  
-Functionalitatea: Defining actions that must execute when the user interacts with those objects graphics application.

The components used to create Swing GUI can be grouped as follows:  
Atomic components JLabel, JButton, JCheckBox, JRadioButton, JToggleButton, JScrollBar, JSlider, JProgressBar, JSeparator  
Complex components JTable, JTree, JComboBox, JSpinner, JList, JFileChooser, JColorChooser, JOptionPane  
JTextField component for editing text, JFormattedTextField, JPasswordField, JTextArea, JEditorPane,  
JMenuBar menus, JMenu, JPopupMenu, JMenuItem, JCheckboxMenuItem, JRadioButtonMenuItem  
Intermediate containers JPanel, JScrollPane, JSplitPane, JTabbedPane, JDesktopPane, JToolBar  
Containers senior JFrame, JDialog, JWindow, JInternalFrame, JApplet

*Classes*

Using UML diagram, above, has been designing, highlightening packages, classes, methods, and the links between them, respecandu the OOP model.  
UML is a visual modeling language, it is still not a programming language because it lacks full support to replace semantic and visual programming languages. Language is for visualization, specification, construction and documentation systems applications, but has limitations in terms of code generation. UML combines the best techniques and practices of engineering progamării that have proven effective in building complex systems.  
Charts are graphs showing the symbols of elements modeling (model element) arranged to illustrate a particular part or a specific aspect of the system. A model usually has several charts of the same type. A diagram is part of a specific view, but there is a possibility to make a chart in multiple views, depending on its content.  
DictionaryProc interface is the interface that will implement the Dictionary class, so all methods that we want to achieve on the dictionary in the Track application must be defined in this interface. Thus any subsequent development will be easy. We realized this interface for each method pre and post conditions to be easier to understand for someone who wants to achieve further development of the application.  
DictionaryProc Dictionary class implements the interface, and Serializable class representing the stock words. This class is implemented with a HashMap that is the key word, of type String, and the value synonym of type String. We implemented as object type Serializable Dictionary will be serialized and deserialized. DictionaryProc some methods that help us realize these methods. All methods herein are used in class, namely the GUI user interface.  
Class aims to achieve GUI graphical interface itself (with Labels, buttons, etc). The interface is graphical user interface based on a display system that uses graphics. The graphical interface is called System-visual graphical display on a screen. To present all the information and actions available, provides a GUI icons and visual indicators as opposed to text-based interfaces, which provides only command name

**GUI**

(-) JFrame Frm

(-) JMenuBar meniu

(-) JMenu Dictionars

(-) JMenu search

(-) JMenu isConsistent

(-) JMenuItem showDictionars

(-)JMenuItem insertDictionar

(-)JMenuItem deleteDictionar

(-)JMenuItem updateDictionar

(-)JMenuItem searchCat

(-)JMenuItem sConsistent

(-)JScorllPane scrollPane

(-)JTable table

(-)JLabel insertWord

(-)JLabel insertSynonym

(-)JLabel inserare

(-)JLabel actualizare

(-)JTextPane actualizareNota

(-)JTextPane consistent

(-)JTextPane consistent Nota

(-)JLabel stergere

(-)JLabel cautare

(-)JTextField insertWordText

(-)JTextField insertSynonymText

(-)JTextField insertNewSynonymText

(-) JButton insertButton

(-) JButton updateButton

(-) JButton deleteButton

(-) JButton searchButton

# Dictionary dictionary

(+)GUI()

(-)i

(+)actionPerformed

#DefaultTableModel buildTableModel

(-) JTable showSynonymus

(-)JTable search: String word

(-)clearScreen

**Dictionary**

(+)HashMap<String,String> dictionary

(+)Dictionary()

(+)readsynonymus()

(+)writesynonymus()

(+)addsynonymus()

(+)updatesynonymus()

(+)deletesynonymus()

(+)isConsistent()

(+)isWellFormed()

**DictionaryProc**

(+) readsynonymus()

(+) writesynonymus()

(+) addsynonymus(String word, String synonym)

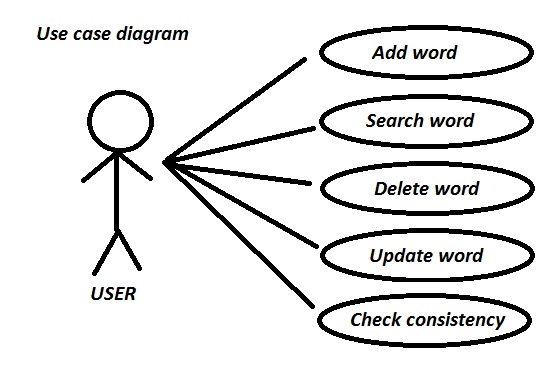
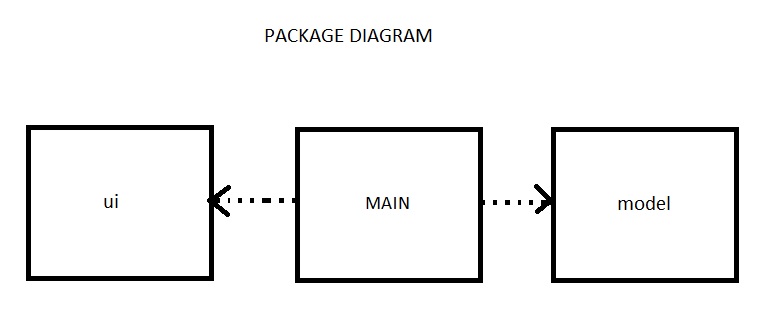
(+) update synonyms()

(+) deletesynonyms(String word)

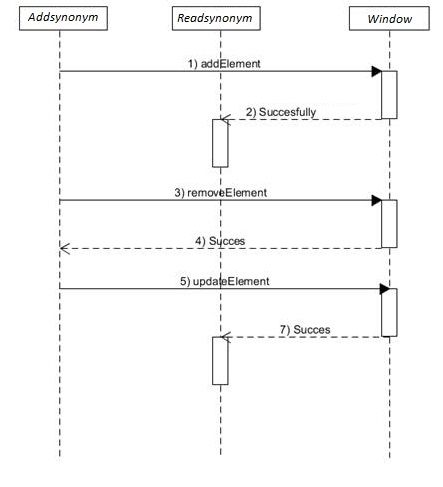
(+) isConsistent()

**Main**

(+)main



*Sequence Diagram*



*Implementation and testing*

Implementation code itself was written after the UML model seen above. Will now be described in detail classes, methods used for this program.  
 Dictionary defines the object type class HashMap <String, String>  
// Restore state object.  
readsynonyms public void ();  
// Save the state object  
writesynonyms public void ();  
// Add new words and synonyms in dictionary  
// W @ param1 String represents the first word  
// @ Param2 is the second word word String  
public void addsysnonyms (word String, String synonym);  
// Update of words from the dictionary  
// W @ param1 String word is to be removed from the dictionary  
// @ Pre condition is checking for word came as a parameter  
// @ Bad condition is hashed size check, it must be equal to the size Previous // minus 1  
public void updatesynonyms (String word, oldSynonym String, String newSynonym);  
// Delete words from the dictionary  
public void deletesysnonyms (String word);  
// A dictionary is consistent when we find synonymous and synonymous word = = // word. For example: sad = angry, and upset = sad "  
public boolean isConsistent ();

*Results*

When the application runs, that too all data from a file. Such application will always keep all data. After I saved all the data already we can perform various operations on bank accounts. After I made the desired operations (deletion, insertion or addition word) all data will be written back into the file, so they keeping their eyes safe. Any surgery performed at runtime application will save a file, so it's like I use a database but use a file where we store everything dictionary. When the object type runs where it is stored dictionary HasMap will retrieve all the data in that file.

*Conclusions & future developments*

The purpose of this theme was touched, so the program runs .Operatiile were implemented as simple, aiming to be an understanding and linearity codului.Cu They certainly are not the only methods but were best understood by me and -They helped to realize the functionality of the application.  
What I learned?  
 First, what is more important is that enriched my knowledge on object-oriented programming, I learned to ask better in practice OOP paradigms. I learned to use concepts such as: Desing by Contract and Serializable and Deserializable. I realized that the first thing on a project, especially at higher proietele is understanding the requirement and putting it into practice by structuring their ideas and analysis.

*Further developments*

Depending on the purpose for which it will use, the application has many possibilities for expansion, development and the improvements both algorithms and the design and operations they perform. Some of these developments can also:  
1. Interface attractive  
2. More synonymous to one word  
3. Users different for certain changes to the dictionary  
These things can be fixed with a few lines of code placed in the right place.